Accuracy and Performance Scaling of LASNEX, a Fortran 77 Radiation-Hydrodynamics Code, on an ASCI Platform (U)

N.M. Hoffman, J.L. Collins, W.J. Powers, J.C. Rieken, C.W. Cranfill, M.R. Clover, D.D. Weeks, and H.K.S. Deaven
Los Alamos National Laboratory

LASNEX (Zimmerman and Kruer, 1975) is a 2D Lagrangian radiation-hydrodynamics code written mainly in Fortran 77. Modifying the 330,000-line code, to allow it to run on the Silicon Graphics Inc. (SGI) Origin 2000 machines at Los Alamos, has been in progress for almost two years, and is now nearly complete. Preliminary studies have been carried out for simple computational problems exercising the hydrodynamics and radiation transport subroutines, to compare the accuracy and speed of the ported code to the Cray J90 version. (U)

Keywords: LASNEX, SGI Origin 2000, radiation hydrodynamics, similarity solutions, parallel processing

Introduction

LASNEX is the principal radiation-hydrodynamics code used at Los Alamos National Laboratory for the design and analysis of laser-driven high-energy-density laboratory experiments and for inertial-confinement-fusion (ICF) research. In the mid 1970's LASNEX was imported to Los Alamos from Lawrence Livermore National Laboratory (LLNL), where it was developed to model the effort to achieve laser-initiated thermonuclear fusion (Zimmerman and Kruer, 1975). Since that time LASNEX has been maintained at Los Alamos by a team of five to ten physicists and computer scientists. Los Alamos LASNEX has been developed independently of LLNL LASNEX. While a specific set of subroutines or a numerical treatment may occasionally be shared between the two laboratories, the codes maintain their own unique identities, and in that way are critical to the lab-to-lab peer review process.

In mid-1996 it became apparent that porting LASNEX to Silicon Graphics Inc. (SGI) computers, running the IRIX operating system on MIPS processors, would be desirable in the future. We expected that the Cray J932 and T94 machines on which the code was then running would be replaced by SGI machines eventually. Our motivation was in fact two-fold; we desired

- to speed up the code through use of the multiprocessor architecture of the SGI Origin 2000s; and
- to create a platform-independent version of the code, regardless of hardware speed.

The goals for the port are thus correspondingly two-fold. We hope

- to speed the code up by a factor of 30 in wall-clock time relative to the Cray J932, so that calculations currently requiring three months of single-processor execution on the J932 will run in three days of multiprocessor execution on the SGI; and
- to have the code running with full functionality on the SGI machines in time to perform adequate testing and develop users' confidence in the SGI version before the Crays are removed.

Porting the code

Work on porting the code began at a low level in November 1996, always subject to the provision that support of the Cray version of the code, on which users depended to do their jobs, not be interrupted. At present (October 1998), much of the physics and input-output functionality of the code has been activated on the SGIs. On the SGIs, the LASNEX development team now routinely carry out 2D calculations with multigroup radiation diffusion and hydrodynamics, employing ideal-gas analytic equations-of-state and analytic opacities. A modular stand-alone package for accurate radiation transport also now runs on the SGI machines. This package was first activated, for purposes of debugging, in a distributed heterogeneous processing mode; that is, the main LASNEX program executed on the Cray J932 while the radiation package executed on multiple processors of an SGI Origin 2000, communicating with the main program via files on an NFS file system. As of October, the main program can run on the SGIs as well.

The SGI code now reads generator files, handling the macro language correctly, and it reads restart files (written by the SGI code). It allows asynchronous terminal input and output, so that parameter values may be queried and changed while the code is running, and the code may be suspended and resumed. The SGI code also writes POP link files (movie dumps), so results can be compared with Cray results using the POP plotting code. (On the SGIs, POP can correctly handle Cray-generated POP files, allowing easy comparison to SGI-generated POP files.)

At present, the following capabilities of the SGI version of the code are being tested: SESAME equation-of-state files, laser rays, non-LTE atomic physics (XSN), and the automatic rezoner. In the near future, the SGI code will be able to read multigroup opacity tables.

Tasks for porting

Porting the code has involved numerous tasks, some of which were foreseen at the outset, others of which were identified as we progressed. One obvious necessity was to replace non-standard Cray-specific Fortran with portable equivalents; examples are Cray conditional vector merges, which we replaced with in-line *if-then-else* statements or with replacement subroutines that provide equivalent functionality. Cray Boolean type is not supported on the SGI, and had to be replaced. Another obvious task was to replace subroutines written in Cray assembler language with Fortran equivalents. A major task was to modify memory manager routines to handle the byte addressing of the IRIX operating system. Various Fortran disk I/O routines were replaced with C equivalents, to take advantage of the faster, block-wise data transfer of the C routines. Numerous examples of incompatibility between SGI and Cray compilers were encountered,

including, for instance, different typing of library functions; some Cray *shift* functions handle real or integer variables interchangeably, but the SGI versions of the functions do not. Some of these incompatibilities amounted to bugs in the compiler, as for example an inconsistency in the way character strings were passed to subroutines, depending on whether the string appeared explicitly as an argument or was contained in a variable. These and other details of how the port was accomplished will be discussed at greater length in a future document.

We considered two competing ways of organizing and scheduling the work for the port. In the functionality-based approach, one would focus on porting a specific physics functionality, such as multigroup radiation diffusion or hydrodynamics, to the new machines. This allows early demonstrations of progress, but requires that porting tasks be repeated many times on different physics packages, with the attendant risk of inefficiencies and errors. In the language-based approach, on the other hand, the code is treated as a single unit, in which all occurrences of a particular language construct are modified at once, and the process is repeated for all required modifications. This is efficient, but delays the demonstration of particular code capabilities. In practice, both approaches were used to some extent. The functionality-based approach played a role at the beginning and near the end of the project, but the language-based approach dominated during most of the work and was probably the mode in which most of the porting modifications were carried out.

While porting did not require it, we occasionally made code modifications to the Cray-specific parts of the code designed to improve the code's portability and conform with current software practice. Thus we converted Hollerith variables to characters in some places, and cleaned up common blocks where Hollerith had been mixed with integers and reals. Other efforts included minimizing integer Boolean operations, and translating Cray multitasking instructions into SGI multitasking instructions.

Test problems

To carry out the initial testing of the SGI version of LASNEX, we are using a simple radiation-hydrodynamics problem with an exact solution. This is the eighth similarity solution found by Coggeshall (1991), for the equations of 1D, one-temperature inviscid hydrodynamics with heat conduction. It describes an expanding isobaric heat-conducting sphere. The solution is

$$\rho(r,t) = \rho_0 r^{\frac{k-1}{\beta-\alpha+4}} t^{-k-1-\frac{k-1}{\beta-\alpha+4}}$$

$$u(r,t) = r/t$$

$$T(r,t) = T_0 r^{\frac{1-k}{\beta-\alpha+4}} t^{(1-\gamma)(k+1)+\frac{k-1}{\beta-\alpha+4}}$$

where ρ, u, and T are density, velocity, and temperature, respectively. The conduction mean free path is given by $\lambda(\rho,T)= \lambda_0 \rho^{\alpha} T^{\beta}$. The variable k is a geometry factor, equal to 2 for spherical geometry, while γ is the ratio of specific heats for an ideal gas. In this solution, pressure $\propto \rho T =$

constant in space. Since there is no pressure gradient, mass points move with constant velocity, as may be verified by considering the Lagrangian derivative of velocity, which vanishes:

$$\frac{D}{Dt}u(r,t) = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial r} = -\frac{r}{t^2} + \frac{r}{t^2} = 0\,.$$

The divergence of the heat flux vanishes as well, as may be seen by evaluating the expression for the flux:

$$F(\rho,T) = \frac{-4ac}{3}\lambda(\rho,T)T^3\nabla T = \frac{-4ac}{3}\lambda_0\rho^{\alpha}T^{\beta+3}\frac{\partial T}{\partial r}\,.$$

Upon inserting the expressions for ρ and T given above we find that $F(r,t) \propto r^{-k}$. Thus the flux's divergence, which is proportional to $\partial(r^k F)/\partial r$, vanishes for all choices of parameters. In our test as implemented within LASNEX, the electrons, ions, and radiation are constrained to be tightly coupled, so the temperatures of all three fields are clamped together. The radiation diffusion package is then used to solve the flux equation.

In our test problem, we choose γ = 5/3, k=2, α=-1, and β=2. Then

$$\rho(r,t) = \rho_0 r^{1/7}\ t^{-22/7}; \tag{1}$$

$$T(r,t) = T_0 r^{-1/7}\ t^{-13/7}; \tag{2}$$

$$u(r,t) = r/t\,. \tag{3}$$

.

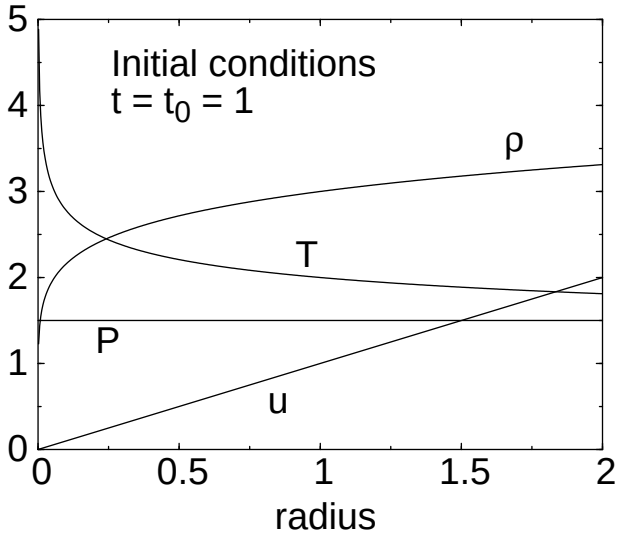

Fig. 1. Variation of density, temperature, pressure, and velocity with radius at initial time in Coggeshall's eighth problem, for the choice of solution parameters described in the text. Evolution of system is self-similar, so shape of these curves is invariant as time increases.

These expressions are graphed in Fig. 1 at a fixed time. The structure shows a hot, low-density core, constant pressure, and linear velocity increase with radius.

The expressions (1) - (3) give the density, temperature, and velocity structure as functions of the Eulerian variables r and t. To evaluate these quantities at a (Lagrangian) mass point, we find the trajectory $R(t)$ of the mass point (assumed to be initially at $r = R_0$ at time $t = t_0$) from

$$\frac{d}{dt}R(t) = u(R, t) = R/t$$

so that

$$dR/R = dt/t$$

resulting in

$$R(t) = R_0 t/t_0 .$$

Inserting this expression in the expressions for density and temperature leads to

$$\rho(R(t), t) = \rho_0 (R_0 t/t_0)^{1/7} \quad t^{-22/7} = \rho_0 (R_0/t_0)^{1/7} \quad t^{-3} \tag{4}$$

$$T(R(t), t) = T_0 (R_0 t/t_0)^{-1/7} \quad t^{-13/7} = T_0 (R_0/t_0)^{-1/7} \quad t^{-2} \tag{5}$$

which demonstrates that the scaled quantities ρt^3 and Tt^2 are functions of the Lagrangian variable R_0, and invariant in time at any mass point. We use this result to test the accuracy of the code, as discussed below. We also use this result to define boundary conditions to apply at the outside of the sphere, which is a Lagrangian point. Equations (4) and (5) show that the pressure P at a Lagrangian point is

$$P(R(t), t) = P_0 t^{-5} \propto \rho(R(t), t) T(R(t), t). \tag{6}$$

In principle the solution [Eqs. (1)-(3)] describes a sphere of infinite radius, but it also describes a sphere of finite radius if the temperature and pressure at the outer boundary of the sphere are given by Eqs. (5) and (6). The boundary conditions then, in a sense, represent the effect of the rest of the infinite sphere which must of course be omitted from any real calculation.

Accuracy

To test the SGI version's accuracy in computing Coggeshall's eighth problem, we assess the degree to which functions of R/t such as ρt^3, Tt^2, and Pt^5 are preserved as invariant during the course of a calculation. If at a certain time we plot ρt^3 as a function of R/t, the curve should be superimposed on other similar plots of ρt^3 done at other times; any deviation represents an inaccuracy of the code. As a quantitative measure of accuracy, we define the relative density error $[\rho(R_0, t_0)t_0^3 - \rho(R_0, t)t^3]/\rho(R_0, t_0)t_0^3$, and examine its value throughout the mesh at various times. Figure 2 shows the relative density error has a value of less than one part per million in the majority of the sphere during the interval $1.3 < t < 4.0$, for a problem beginning at $t_0 = 1$. (For this and the following two figures, the Cray code gives results very similar to the SGI code's results.)

Figure 2 shows that the relative density error increases towards the origin ($R/t = 0$) and towards the outer boundary ($R/t = 2$) of the problem. At the outer boundary, the error is presumably related to inaccuracy in the boundary conditions, which are specified in the code input file as

tables of times and corresponding values of temperature and pressure. Such tabular, piecewise-linear input is only an approximation to the continuous variation of these quantities necessary to give exact agreement with the similarity solution near the boundary, so it is to be expected that some error occurs there.

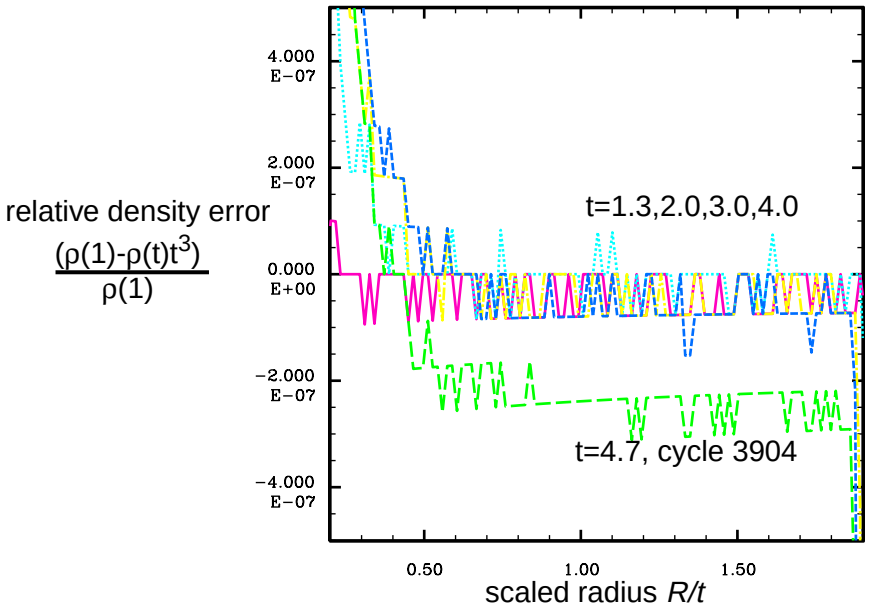

Fig. 2. Relative density error as a function of scaled radius, for several times throughout the calculation. The magnitude is less than 1 part per million over most of the mesh ($0.2 < R/t < 1.8$) for these times.

The error at the origin is more problematic. In Fig. 3 we plot the scaled pressure $P(t)t^5/P(t_0)t_0^5$ as a function of R/t at various times, over the range $0 < R/t < 0.5$. Ideally, all curves should coincide with the value 1.0 for all radii and times. But errors arise quite early at the origin and propagate outward, ultimately approaching asymptotically a value of 0.02% or less throughout the mesh.

The error in other physical fields is comparable near the origin. Figure 4 shows a plot of the scaled density $\rho(t)t^3/\rho(t_0)t_0^3$ at various times. Again, departures from 1.0 indicate error. Here it is clear that the largest error occurs at early time, and generates a propagating pulse that moves outward, while the error at the origin oscillates and decays. Although these errors are not large, it is worth pursuing their cause in order to identify the effect in other situations. The behavior seen here suggests some problem associated with the beginning of the calculation, and restricted to the zone at the origin. This may simply be an inaccuracy in the initial conditions. For example, perhaps the initial density in the central zone should be chosen to give the correct mean density in the zone, rather than simply evaluating Eq. (1) at the zone's midpoint, as is currently done.

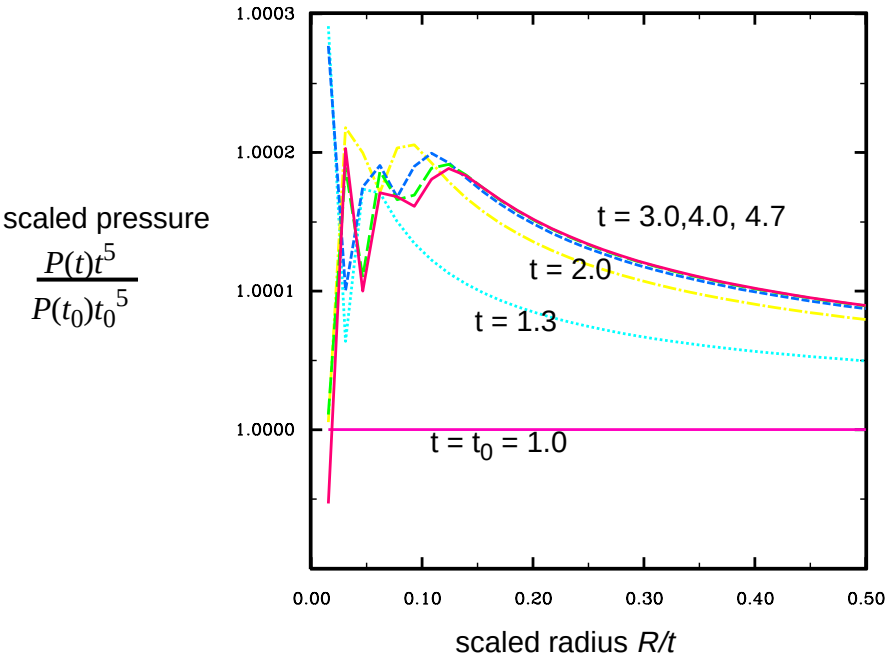

Fig. 3. Scaled pressure as a function of scaled radius, for several times throughout the calculation. Ideally all curves should be superimposed with the value 1.0 everywhere. Error arises at the origin, propagates outward, and ultimately saturates.

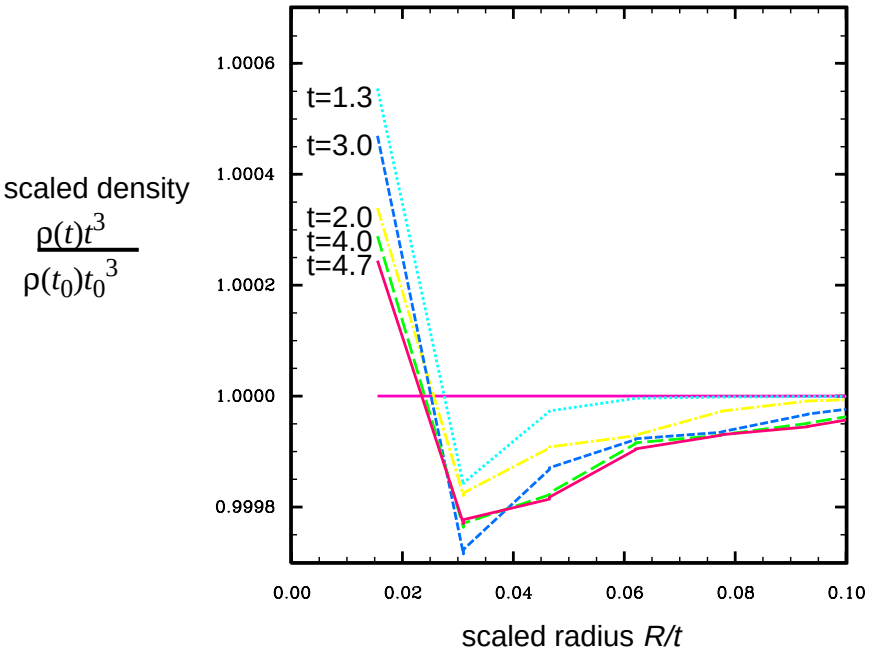

Fig. 4. Scaled density as a function of scaled radius, for several times throughout the calculation. Ideally all curves should be superimposed with the value 1.0 everywhere. Error arises at the origin at early time, propagates outward, while oscillating and decaying near origin.

Performance

Single-processor performance. To gather data about the calculational speed of the ported version of LASNEX on the SGI machines, we performed a series of calculations of Coggeshall's eighth problem, in which the various calculations had differing numbers of zones in the computational mesh. Calculations were run on both the SGI Origin 2000 machine "k01" (which uses 250 MHz processors) and on the Cray J932 machine "zeta", and the relative elapsed wall-clock time was compared. Care was taken to compensate for the demand on the processors from competing jobs, based on the reported fraction of processor utilization during each run. Each calculation was run for 4000 cycles, used 100 photon energy groups, and used the partial-temperature technique for multigroup radiation diffusion.With these parameters, the code spends 80% to 90% of its time in the radiation diffusion package.

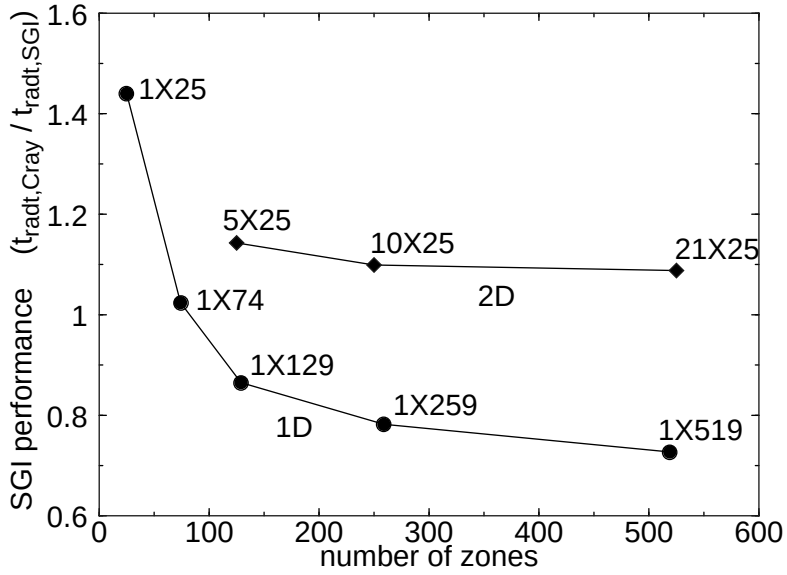

Fig. 5. Speed of single-processor SGI Origin 2000 computation relative to single-processor Cray J932 computation, defined as ratio of time spent in radiation diffusion on Cray to that on SGI. This quantity is plotted as function of problem size, i.e., number of zones in mesh.

Figure 5 shows the relative speed of the Origin 2000, running unoptimized code, to the Cray J932, running optimized vector code, as a function of the number of zones in the mesh, for several calculations carried out on single processors of each machine. The quantity plotted on the vertical axis is the ratio of the effective (scaled to 100% processor utilization) wall-clock time spent in the radiation transport routines. Because the Cray was running optimized code and the SGI was not, the comparison is biased in favor of the Cray, but is an accurate picture of the relative performance at the time of this writing. As the code undergoes further testing and debugging on the SGI, we will eventually compile it with optimization, and presumably improve its performance. The figure shows that very small problems run faster on the SGI than on the Cray, but as problem

size increases, the vectorization of the Cray code becomes progressively more important and the relative performance of the SGI worsens. Still, the SGI speed is 70% that of the Cray even for quite large 1D meshes. For 2D meshes that are not too large, as shown, vectorization is not so dominant an advantage and the SGI speed exceeds that of the Cray. These results are not an exhaustive study of all possible mesh sizes, which is not warranted at this stage in the development of the SGI code anyway, but they do indicate that the single-processor SGI performance experienced by users will be roughly comparable to the Cray J932.

Multiple-processor performance. The multigroup diffusion package has recently been parallelized on both the Cray and SGI platforms, using Cray multitasking routines on the Cray and the OpenMP library on the SGI. A series of calculations was performed to assess the speedup realized from multiprocessing on the SGI, using 1D and 2D versions of Coggeshall's eighth problem. For these calculations, it was necessary to perform implicit multigroup diffusion, because it is that capability that has been multitasked on the SGI. Calculations were run for 2000 cycles, using 100 photon energy groups.

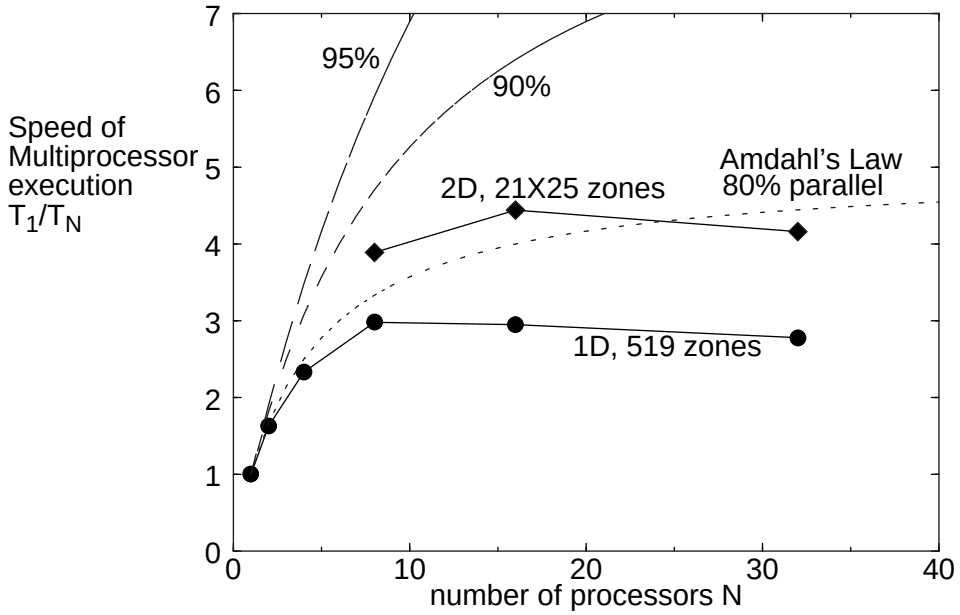

Fig. 6. Speedup of SGI Origin 2000 computation resulting from multitasking of radiation diffusion. Dashed curves are evaluations of Amdahl's Law, for various fractions of code parallelism 1-β as labeled.

Results of multiprocessor tests are shown in Fig. 6. In general, the speedup (shown as the ratio of T_1, the effective wall-clock time using 1 processor, to T_N, the effective wall-clock time using N processors) increases for a few processors but then saturates and even decreases, for more than 8 to 16 processors. These curves are examples of behavior that is approximately predicted by Amdahl's Law, which relates T_N to T_1 as follows:

$$T_N = \beta T_1 + (1-\beta)T_1/N.$$

Here β is the fraction of the code that runs serially, while 1-β is the fraction that runs in parallel and therefore can be speeded up by parallelization. As *N*, the number of processors, increases to infinity, the speedup T_1/T_N can never increase beyond 1/β. So there is a premium on reducing the amount of serial code.

Amdahl's Law in this form, however, does not account for the drop-off in performance as *N* increases beyond about 16 in Fig. 6. Such a drop-off is presumably due to the communications overhead incurred as more processors are added. We can generalize Amdahl's Law to account for this effect by adding a term that causes T_N to increase with increasing *N*. An example is

$$T_N = \beta T_1 + (1-\beta)T_1/N + CT_1N^{\alpha} \quad .$$

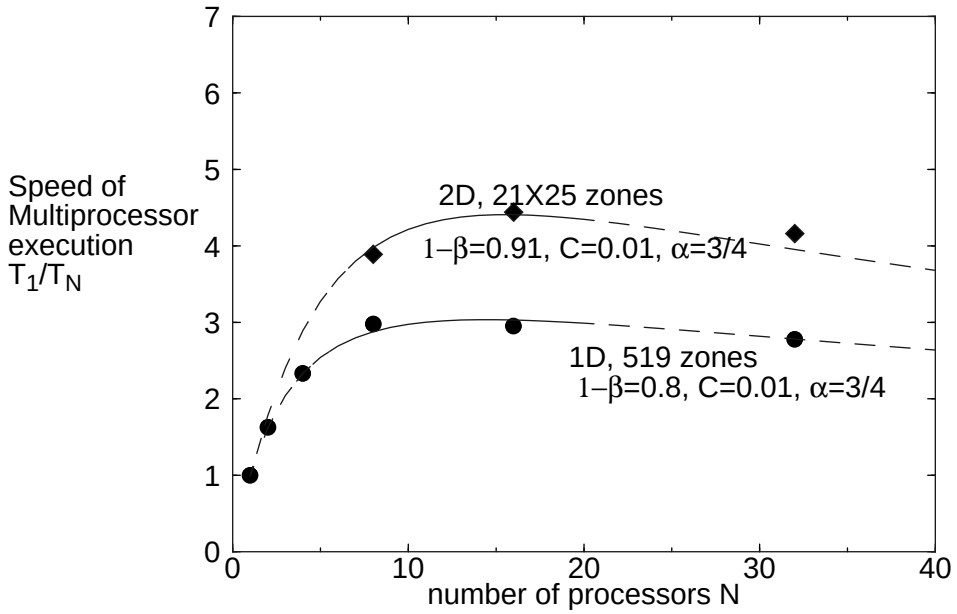

Fig. 7. Speedup of SGI Origin 2000 computation resulting from multitasking of radiation diffusion. Dashed curves are evaluations of a generalized form of Amdahl's Law, intended to account for communications overhead resulting from adding processors. The curves are given by

$$T_1/T_N = \frac{1}{\beta + (1-\beta)/N + CN^{\alpha}}.$$

where the third term reflects the assumption that the communications overhead is proportional to the problem size (measured by T_1) and some power α of the total number of processors. Figure 7 shows the data from Fig. 6 compared to curves given by the generalized form of Amdahl's Law, where the parameters were chosen to give a crude fit to the data. The performance drop-off,

according to this model, results from a rather small (C = 0.01) communications overhead. Speeding up the code thus puts a premium on reducing such overhead, as well as reducing serialism.

Conclusion

The task of porting LASNEX to the SGI Origin 2000 system is still in progress, but is beginning to bear fruit in the sense that some initial capability is now available on the SGIs. Multigroup radiation diffusion with hydrodynamics, using analytic equations of state and opacities, is functional. Capabilities now undergoing testing include SESAME equation-of-state files, laser rays, non-LTE atomic physics (XSN), accurate radiation transport, and the automatic rezoner. Studies of code accuracy are in progress, using comparisons to analytic similarity solutions. Parallel multigroup diffusion on the SGI allows a speed-up in execution of about 4.5, using 16 processors, indicating that more work is necessary to increase parallelism and reduce communications overhead.

References

Coggeshall, S.V., *Phys. Fluids* A **3,** 757 (1991).
Zimmerman, G.B. and Kruer, W.L., *Comments Plas. Phys. Controlled Fusion* **2**, 51 (1975).

LA-UR- 98 - 5712

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

TITLE: **Accuracy and Performance Scaling of LASNEX, a Fortran 77 Radiation-Hydrodynamics Code, on an ASCI Platform**

AUTHOR(S): **N.M. Hoffman, J.L. Collins, W.J. Powers, J.C. Rieken, C.W. Cranfill, M.R. Clover, D.D. Weeks, and H.K.S. Deaven**

SUBMITTED TO: Proceedings of 1998 NECDC, Las Vegas, NV 26 - 30 October 1998

Los Alamos **Los Alamos National Laboratory**
Los Alamos, New Mexico 87545